AD-A049 486   ARMY ELECTRONICS COMMAND FORT MONMOUTH N J   F/G 9/2

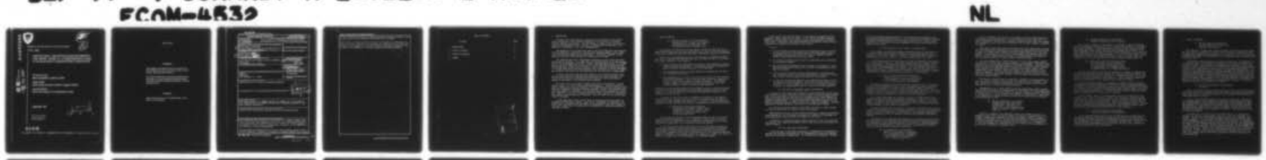
COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE FINAL REPORT. --ETC(U)

SEP 77 T CONRAD, R ESTELL, L HAYNES

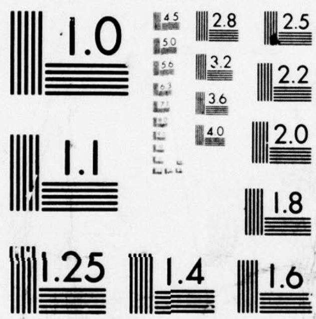
UNCLASSIFIED   ECOM-4532   NL

| OF |

AD
A049486



END
DATE
FILMED
3-78
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Research and Development Technical Report

ECOM - 4532

COMPUTER FAMILY ARCHITECTURE SELECTION COMMITTEE - FINAL REPORT, VOLUME IX - A CONSIDERATION OF ISSUES IN THE SELECTION OF A COMPUTER FAMILY ARCHITECTURE

Thomas Conrad
Naval Underwater Systems Center

Robert Estell
Fleet Combat Direction Systems Support Activity

Leonard Haynes
Naval Ship Research & Development Center

September 1977

DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

ECOM

US ARMY ELECTRONICS COMMAND FORT MONMOUTH, NEW JERSEY 07703

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The citation of trade names and names of manufacturers in this report is not to be construed as official Government indorsement or approval of commercial products or services referenced herein.

Disposition

Destroy this report when it is no longer needed. Do not return it to the originator.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
14 ECOM-4532

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

9 Research and development technical rept.,

4. TITLE (and Subtitle)
6 Computer Family Architecture Selection Committee Final Report. Volume IX. A Consideration of Issues in the Selection of a Computer Family Architecture.

5. TYPE OF REPORT & PERIOD COVERED

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
10 Thomas Conrad, [redacted]
Robert Estell [redacted]
Leonard Haynes [redacted]

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Center For Tactical Computer Services (CENTACS)
DRSEL-NL-BC
Fort Monmouth, N.J. 07703

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
16 1L7_62701_AH92~~8109~~
17 B1

11. CONTROLLING OFFICE NAME AND ADDRESS

12. REPORT DATE
11 Sep~~tember 1~~977

13. NUMBER OF PAGES
19 Pages 12 23p.

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
CENTACS
DRSEL-NL-BC
Fort Monmouth, N.J. 07703

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES
This report consists of a summary and nine (9) volumes. It is the result of joint Army/Navy work. The complete report may be separately published by the Naval Research Laboratories.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Computer Family Architecture, Evaluation Criteria, Microarchitecture

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
This volume describes various issues confronted by the Computer Family Architecture (CFA) Selection Committee in its search for a standard military computer architecture. This discussion of issues reveals the difficult nature of the Committee's task. Further, it demonstrates the sources of the Committee's confidence in its final recommendations.

The motivation for including this data in the final Committee report derives from the Committee's experience in bringing new Committee members "up to speed"

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

037 620 JOB

20\. and in describing progress to non-participants. Invariably, newcomers raised questions with which the Committee had already dealt at length.

Eleven issues of premise and thirteen issues of procedure are discussed in detail. Each issue is presented as a question. Its significance is discussed and the impact of the Committee's decision relevant to the issue is described. The decision of the Committee is stated and supporting rationale provided.

TABLE OF CONTENTS

SECTION	PAGE
1. INTRODUCTION	1
2. ISSUES OF PREMISE	2
3. ISSUES OF PROCEDURE	7
4. SUMMARY	16

1. INTRODUCTION

The pages that follow describe various issues confronted by the Computer Family Architecture (CFA) Selection Committee in its search for a standard military computer architecture. This discussion of issues reveals the difficult nature of the Committee's task. Further, it demonstrates the sources of the Committee's confidence in its final recommendations.

The motivation for including this data in the final Committee report derives from the Committee's experience in bringing new Committee members "up to speed" and in describing progress to non-participants. Invariably, newcomers raised questions with which the Committee had already dealt at length.

The Committee believes a consolidated presentation of the issues and their disposition is valuable. It is hoped thus to eliminate doubts as to the thoroughness of the investigation conducted. Further, the Committee believes it is worthwhile to point out where the constraints of limited time and resources may have limited the depth to which certain aspects of the investigation could be pursued. It is hoped that others with the opportunity and inclination to extend the Committee's analysis can profit from its experience and find direction in these pages.

It has already been noted that the attention paid to many pertinent issues was not always obvious in the decisions rendered or the results reported. It should also be noted that Committee decisions required a two-thirds majority support of the members. Few decisions were unanimous.

Of course, some issues were of greater significance than others. We have found it useful to classify them in two broad categories. Several recurrent issues have dealt with the basic premises of the Computer Family Architecture effort. These are fundamental questions of overall approach. Other issues developed over particular aspects of procedure. The evaluation process was necessarily a serial one. Naturally, in such a candidate screening process, results are a function of the sequence in which measures are applied. This fact gave rise to various questions of procedure.

In the pages that follow, eleven issues of premise and thirteen issues of procedure are discussed in detail. Each issue is presented as a question. Its significance is discussed and the impact of the Committee's decision relevant to the issue is described. The decision of the Committee is stated and supporting rationale provided.

2. ISSUES OF PREMISE

(1) Why was the original list of candidates limited to nine? In particulary, why did the Committee not consider any of the architectures of CDC, Honeywell, etc.?

This question is significant for two reasons. Aside from the fact that a vendor whose products were not evaluated might object, there is the more important issue that the Committee might have overlooked a potentially superior architecture.

Fifteen Army and twenty-two Navy users and developers of computing systems were asked to nominate architectures which they considered worthy of investigation. This process resulted in the following:

1) Before selecting one architecture to nominate, each respondent informally evaluated several architectures, including some ultimately nominated by others, and many not mentioned at all to the Committee.

2) Each respondent was able to provide technical information on the candidate architecture nominated.

3) Each respondent was expected to continue the evaluation of the candidate architecture nominated. (These last two items allowed the Committee to pursue its task with a minimum of external resources. The alternate proposal, namely, to evaluate all current architectures, would have demanded technical resources external to the Committee.)

Additionally, many Committee members felt (although it was not a formal selection criterion) that the new standard architecture should directly support the ASCII character set. This weighed against architectures whose word lengths were not multiples of eight (8) bits.

In summary, it can be said that a fundamental assumption of the Computer Family Architecture selection process was that if an architecture was worthy of consideration, then someone would nominate it. Altogether, nine candidates were proposed. No limit was set on the number of nominations that would be considered.

(2) Should not tactical data processing requirements be analyzed in detail before attempting to describe a computer architecture which satisfies tactical data processing requirements?

This question was raised by several Committee members at the first meeting of the CFA Selection Committee. It is significant in that failure to analyze requirements first leaves any decision rendered by the Committee open to objection on the grounds that it is a solution to the wrong problem.

It was decided after some discussion at the first CFA Selection Committee meeting that available, unclassified reports covering application requirements for the Navy's All Application Digital Computer and the Army's Fourth Generation Tactical Computer Family would be distributed to all CFA Selection Committee members. Ultimately, six documents ([AlexB 73], [ButlW 72], [HopkJ 75], [KilpP 73], [ScieC 72], [ApgaJ 75]) were distributed to all Committee members.

It was decided that further analysis of requirements was beyond the CFA Committee's scope, resources, and needs. It was left to individual Committee members to draw upon their own varied experience, supplemented by the reports cited above. Consequently, the Committee developed evaluation criteria and test programs which characterize the needs of the military.

In brief:

1) The CFA Committee included cognizant representatives from diverse Army and Navy activities, including the Navy's User Requirements Advisory Group (URAG) and the Army's Center for Tactical Computer Sciences (CENTACS) Requirements Advisory Group (CRAG).

2) Six studies of tactical data processing requirements were disseminated to all Committee members.

3) Most operational requirements relate to speed, size, weight, or other physical characteristics which are a function of implementation rather than architecture.

4) The CFA Selection process stressed characteristics of the widest and most important tactical application requirements as reflected in evaluation criteria and test programs developed for this purpose.

5) In addition, the Army and Navy have each commissioned studies of tactical data processing requirements. The results of these studies will be available to, and will impact the efforts of, the contractor selected to develop an implementation plan for a family of military computers.

(3) *Why select an existing computer architecture?*

To develop a fully mature data processing system architecture and the attendant software could well represent a sizable fraction of the DOD computer budget. Even if this were feasible, the return on investment would be small because experience indicates that most military data processing needs can be satisfied by commercial architectures. It should also be noted here that selection of a commercial architecture does not imply acquisition of commercial hardware. Machines supporting a given architecture can be built to military specification.

By picking an existing computer architecture, deficiencies should already be known and understood, and patches already incorporated.

Nonmilitarized versions of the selected architecture, compatible peripherals, and documentation, can be procured from commercial vendors, and programmers already experienced on the selected architecture are available.

In addition, a ready-made support software base will exist for an existing architecture, thus reducing development and/or acquisition costs associated with support software.

(4) Why select one architecture?

It must be kept in mind that architecture is independent of implementation, and that the selected architecture will be implemented on a family of machines spanning a wide range of capabilities. By selecting one architecture, software

will be transportable between machines, with low end machines trapping on and interpreting unimplemented instructions. By selecting one architecture, computer to computer interface problems will be greatly reduced, and it is more likely that one set of peripherals and device dependent IO routines can be used by all members of the family.

(5) Why were there so few "users" on the Committee?

This question is significant because the current inputs of real users are necessary to insure that technical talents and resources are applied to the solutions of the correct problems. Solutions to pseudo-problems, no matter how sophisticated, are of no benefit to the Army and Navy units in the field.

The Commitee was well represented by real users as evidenced by the membership roster found in Volume I. Some other organizations were invited, but chose not to participate. The Committee recognizes that wider participation would have increased the confidence in the process, if not the quality of the product also. Thus, one of the reasons for producing this report in this form (i.e., including this section of introspective critique) is to allow others to continue the evaluation, without having to repeat the process in its entirety.

(6) Isn't it more appropriate to develop an enhanced version of existing standard Army or Navy computers rather than standardize on an existing commercial architecture?

This question challenges the CFA Commitee's reason for being. The question has been voiced frequently by those who are concerned about large investments in application software for today's military computers, notably the AN/UYK-7, AN/UYK-20, and AN/GYK-12. The CFA Committee did not spend any time in self-justification. Instead, it accepted its charter and proceeded with the task of identifying an optimum architecture for a familiy of military computers.

It is worth noting that the military is at least a decade behind industry in applying computer software and hardware technology. It would be a mistake to adhere so rigidly to current designs that this gap never closes, or even worse, widens. The point is well taken that the military must be prepared to keep pace with developments in the technical marketplace.

It should be noted that the AN/GYK-12, AN/UYK-20, and AN/UYK-7 were actively considered as candidates for the standard architecture. The fact that they placed fourth, eighth, and ninth among nine candidates on the seventeen quantitative measures developed by the CFA Committee indicates that substantial upgrading is required.

The Committee also understood the importance of capturing the existing software investment. It did not feel constrained to act on this requirement because it understood that the implementation plan for the family of military computers would provide implementations in high performance microprogrammable emulators which can execute existing software in real time.

(7) Why standardize on architecture? Wouldn't it be better to standardize on a microprogram level, implementing a "soft" machine with an alterable native instruction repertoire?

There is nothing in the charter or philosophy of the CFA Committee which is inconsistent with standardizing on a microarchitecture at some later time. If future developments indicate that a standard microarchitecture should be selected, this new machine could replace old CFA machines with no other changes in hardware or software.

Much less is known about microarchitectures than about conventional architecture, and attempting to standardize on an advanced microarchitecture carries with it some risk of developing a machine with many elegant features and a few gross mistakes. Currently, there is little or no standardization among military computers. The most common interface between the system developer and computer is the architecture, and this is what we should standardize first.

The very changeability of the microcode will tend to limit common software and tend to proliferate local inventions wasteful of programmers and machine time. Software that is interchangeable must run with common microcode. Users must first obtain the microcode, then the software to build up a library of useful programs. The CFA project, by standardizing on a particular architecture family, also standardizes on the compilers, operating systems, and system software for that family of computers. This gives the user community an extensive initial base upon which to build applications software, and discourages proliferation of duplicate tool building efforts.

The existing software base for dynamically changeable microcoded machines is much less extensive than for machines that have been extant for a decade. Likewise, the pool of programmers, peripherals, and nonmilitarized versions may be much smaller than for one of the selected machines.

Standardization of a "soft" machine at the microarchitecture levels carries the risk that some aspects of implementation of the machine are being standardized. Implementation is the most likely aspect of computer hardware to change drastically in the next decade. The CFA approach (implementation independence) is more likely to be compatible with hardware advances in the next ten years than the standard microarchitecture of 1976.

(8) Why not standardize on an assembly language level rather than on an architecture? This would allow flexibility in selecting word size, number of registers, address field size, etc.

Compilers, assemblers, and operating systems are extremely sensitive to parameters such as word size and the number of registers. Compatibility at the assembly language level guarantees only that the program written in assembly language will assemble as it is carried from one machine to a different machine. It does not guarantee that the program will work correctly if, for example, it is compiling programs for a machine with one size address field, and moved to a machine with a different size address field.

(9) Why not standardize on a high level language or operating system interface?

Again, there is nothing in the charter or philosophy of the CFA Committee which is inconsistent with standardizing on a high level language or on an operating system interface. In both these cases there are several serious problems which would need further study, and where gross mistakes could be made. If we first standardize on an existing, tried and proven architecture, it will make the other tasks easier for later implementation. It is far from clear that a single higher level language can be made to adequately serve all or most higher level language, and, it is not clear what that single standard language should be.

(10) Won't the selected architecture rapidly become obsolete? Why not standardize on a more modern architecture such as an intermediate level language machine, a stacked machine, or a tagged architecture?

Over the past decade or two the vast majority of increases in computer performance have come about through advances in technology and hardware design features and not through changes in architecture. Hardware advances, such as cache memory, instruction look-ahead, pipelining, etc., are architecture independent and can be incorporated into a computer without affecting software.

Because of enormous commercial investments in software, most architecture evolution that has occurred has been in the form of upwards compatible extensions or additions to existing instruction capabilities, which do not affect the ability of a computer to run already existing programs (these programs simply do not make use of the extended operations). Existing commercial software almost guarantees that this pattern will, for the most part, continue.

The CFA concept, then, is not incompatible with future extensions to the instruction set which, for example, support stack operations, operating system functions, etc. However, these additions can be considered only after they have been proven through test and experience. Jumping into new or radical architectures is a risk that the military should leave to the computer industry. Waiting for such undefined new architectures which are "just around the corner" would only result in that many years further delay in standardization.

(11) Was each architecture investigated with equal competence and thoroughness?

To the greatest extent feasible, each architecture was considered equally. The metrics developed to compare architectures were carefully scrutinized by the Committee to insure that they were meaningful measures which did not penalize one architecture over another for an unimportant characteristic. Professor Harold Stone was hired specifically as an auditor to insure that criteria were correctly evaluated for each architecture.

3. ISSUES OF PROCEDURE

(1) Why were some of the evaluation criteria modified for some of the candidate architectures?

This question is significant in that evaluation criteria were used to winnow the large field of candidates, originally nine (9), down to a number that could be extensively measured by test programs, e.g., about three (3). In this process, a candidate that might have benchmarked very well could have been eliminated before testing was done.

The selection process evolved as the project progressed. The Committee felt it altogether proper to learn from its experiences, and to make mid-course corrections as needed. The judgment of the Committee was that some of the architectures were clearly superior to others; and some efforts were made to explore these in depth, even at the expense of writing off the less promising candidates. In particular, the Interdata 8/32 was originally regarded by the Committee as having failed the requirement for complete recoverability from any interrupt. Also, its extended precision floating point capability was available but not yet delivered (a Committee requirement). However, since Interdata indicated a willingness and ability to address these problems and since the Interdata 8/32 consistently fared better than any other architecture on the quantitative criteria, the Committee decided to include that architecture as a finalist.

(2) Why were these specific Test Programs selected?

This is significant because on any given type of test, one architecture will look better than another, and the results will tend to vary with the type of test. Hence, the choice of Test Programs tends to bias the selection from one kind of architecture towards another.

First, a Test Program Subcommittee evaluated proposed tests, submitted by all Committee members, each of whom was responsible for nominating tests that were representative of the characteristics and needs of his systems. Secondly, the Committee as a whole then cast a weighted ballot, which selected the test programs actually coded and run. Finally, the number of tests was limited by the resources available to the Committee for testing -- time and money, as well as skilled programmers.

A related issue is that no very large tests were run. It just was not practical to code very large programs, except in a high level language, e.g., FORTRAN. Such tests would have been valuable measures of the candidate support software systems, i.e., they would have measured the interaction of architectures and captured support software, yielding a mixed measure. The judgement of the majority of the Committee was that the charter was to evaluate architectures, not support software systems. Moreover FORTRAN and COBOL, the only common languages available on the three final candidate architectures, are not widely used in tactical systems. Hence, such a mixed measure would have not helped the evaluation. It would also have been very difficult to define since it is unclear, for example, how to insure that "equivalent" compilers were used with each architecture.

(3) Why wasn't tactical software capture considered the most important criterion in evaluating computer architectures?

This question was dealt with several times by the CFA Committee. Its significance lies in the inference that the first screening process should order the candidates according to capture of application software.

It should be understood that the Committee had enough time and other resources for a detailed comparative analysis of three "finalist" architectures. Hence an initial screening process was required to select three from among the nine under consideration. The Committee decided on an initial screening procedure involving nine "absolute criteria" and seventeen "quantitative criteria" relating to technical merit of the architecture (address space size, I/O overhead, virtualizability, etc.) rather than criteria relating to present investment in hardware or software.

Clearly, if capture of software investment were used as the screening procedure, a different set of finalists would have been analyzed, since there is no significant tactical software capture with the actual finalists, IBM S/370, PDP-11, or Interdata 8/32. Figure 1 illustrates the selection process as actually implemented. In the figure candidate architectures are shown emerging from each evaluation box in rank order.

Two grounds may be used to justify the Committee's decision to screen the candidates on a technical rather than an investment basis. In the first place, the CFA Committee was well aware of the requirement specified in the implementation plan contract that current military processors be emulated in real time by the family of military computers based on the architecture the Committee was to recommend. The capture of application software seemed assured regardless of the CFA Committee's selection. Thus, a screening process based on other grounds seemed appropriate.

It is also true that hard data relating to application software investment is difficult to obtain. Given the limited resources available, the quantitative measures used were a more practical choice. A case might be made for including application software capture as one of the quantitative criteria and using the best estimates available. In the overall score it is not likely that any of the military machines would have benefited sufficiently to make a difference. Possibly the AN/GYK-12 would have made it to the finalists on the basis of such a quantitative measure score. However, the AN/GYK-12 clearly failed to meet the absolute requirement for floating point precision.

(4) Why was the <u>quality</u> of captured software not measured?

The <u>value</u> of captured software was a key item in the selection of a candidate architecture; and value is surely related in some way to quality, as well as quantity.

Quality of captured software is in a real way a subjective question. Apart from the question of whether or not given software works, there remain real questions as to which types of support software are most useful in any given application. Consider, for example, the continuing debate over on-line versus batch programming, over high level languages versus assembly languages,

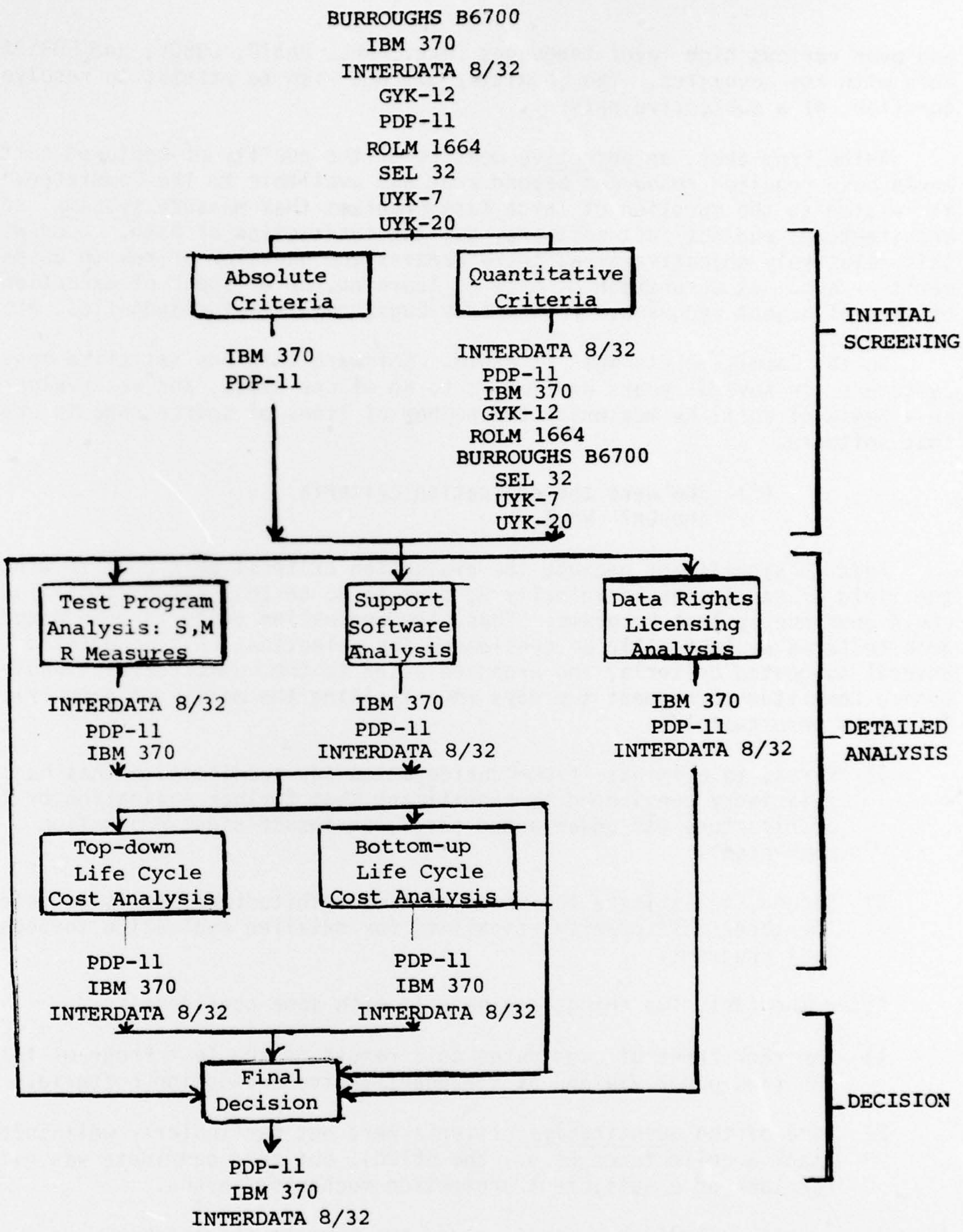


Figure 1. Overview of the Selection Process

and over various high level languages (e.g., APL, BASIC, COBOL, and FORTRAN) each with its advocates. The Committee did not wish to attempt to resolve questions of a subjective nature.

Aside from that, an objective measure of the quality of captured software would have required resources beyond what was available to the Committee. This is related to the question of large Test Programs that measure systems, not just architecture, and not just software, but the interaction of both. Even within this relatively objective area, there remains the question of how to assess merit -- e.g., as a function of ease of learning, or of speed of execution, or of size of object programs, freedom from bugs, quality of diagnostics, etc.

So the Committee did what it could. Software that has satisfied many customers for several years was judged to be of one class, and was evaluated on a basis of cost, by measuring the number of lines of source code to create that software.

(5) How were the evaluation criteria chosen? Why?

This is significant because the evaluation criteria were used to winnow the field of candidates, originally 9, down to 3, so that extensive evaluation could continue by Test Programs. Thus, the evaluation criteria determined which architectures would finally be considered for selection. A subcommittee studied several suggested criteria, and proposed a set to the Committee as a whole. The entire Committee then spent two days investigating the merits of each criterion. The goals were twofold:

1) First, to eliminate from consideration any architecture that had a deficiency considered so significant that further evaluation of the architecture was unwarranted (e.g., an insufficient protection mechanism).

2) Second, to estimate the power of each architecture, so as to select the three most powerful finalists for detailed evaluation through Test Programs.

After the fact, two things can be said with some confidence:

1) The rank order of candidates as a result of the Test Programs followed the rank order yielded by the quantitative evaluation criteria.

2) Some of the quantitative criteria were not particularly well suited to stack architectures (e.g., the B6700), but that candidate was eliminated for lack of a sufficient protection mechanism anyhow.

In summary, the evaluation criteria aided the Committee's thinking and investigation, but did not make the final selection.

(6) Were the most appropriate statistical models used?

Statistical procedures are always controversial. The CFA Committee struggled with the question of an appropriate statistical approach when it conducted its

analysis of the quantitative measures used for screening the candidate architectures. Since the results of the analysis played a part in determining which candidates would be finalists, and since different approaches provided different rankings of the candidates, the choice of statistical procedure was significant.

The question arises in the computation of the overall score on the seventeen quantitative measures. For each measure, scores of all architectures were normalized to have a mean of unity. In addition, five measures (all dealing with address space size in bits) were further normalized for a variance of unity. The Committee determined that such normalizations corresponded to its intuition with regard to the significance of each measure. Proposals to apply square root transformations, logarithmic transformations, or other such filters to the raw data were rejected on the grounds that no reasonable basis could be found for choosing one such transformation over any other. The Committee then determined appropriate weights to be applied to each measure. The procedure for this involved members allocating weights according to what they felt best represented their class of problems. The overall score for an architecture was computed by summing the weighted normalized scores on each measure. In this computation reciprocals of scores were used rather than the scores themselves when necessary to convert all measures to maximization functions (i.e., so that larger score means better performance). Details of this scoring procedure may be found in Volume II.

After this methodology was applied, several alternate procedures were proposed, [PalmB 76], [ThomM 76]. In particular, one procedure called for using negative weights rather than score inversion for measures which score lower values for better performance. Another method proposed applying threshold functions to avoid the necessity for normalization.

The Committee decided that these alternate methods were reasonable methods for computing scores but found no evidence that they provide a clearly superior or more rigorous treatment of the problem. Consequently, the Committee decided to use the original procedure which, at least, was conceived prior to seeing any results of applying other procedures.

Some may claim the results of the CFA Committee's work are meaningless in that they stem from a somewhat arbitrary choice of computational methods. However, the Committee was constrained by a need for a timely selection of a set of finalist architectures for detailed study. Having chosen a reasonable computational method, the Committee chose to abide by the results it offered rather than conduct a long and possibly inconclusive study of alternate methods for screening the candidates.

As for the robustness of the results achieved, Dr. Harold Stone (University of Massachusetts) audited the data and analyzed the effects of the weighting of the measures, their normalization, and the measures themselves in the selection of finalists [StonH 76]. His analysis, which is included in Volume II, concluded that for the given weights, the finalists selected are very insensitive to the exact details of the selection procedure. That is, "almost any reasonable methodology for measuring the key concepts quantitatively would select the same finalists....The selection of three finalists was a function of the preponderance of the heavily weighted data in their favor, rather than due to idiosyncrasies or biases in the measuring and weighting process."

Finally, it should be noted that the quantitative measures were used only to select one of the three finalists. The other two (S/370 and PDP-11) were chosen

because only they unconditionally passed all the absolute requirements set by the Committee. The final quantitative scores for the S370 and PDP-11 were not known to the Committee until after the Committee resolution to accept as finalists those architectures passing all the absolute criteria.

Statistical design was also important in the test program evaluation of the three finalist architectures. The Committee dealt more rigorously with statistical issues in this case. Dr. Paul Shaman (Carnegie-Mellon University) provided direction of the experimental design for the Test Program analysis. Further, he provided an interpretation of the statistical significance of the results. The statistical design and interpretation of the test program experiment is discussed at length in Volume III.

(7) Are we really going to capture any support software? Don't available support software programs require a particular operating system, and aren't there minor differences in architecture between the various members of the commercial architecture families?

In order to run most support software, a given operating system will be required. This, however, causes no significant problem. Since the manufacturers of the candidate architectures do support the operating systems which are required to run their support software, Government support for these operating systems should be minimal. Stated in other words, a captured program can be thought of as a load module consisting of a support program and operating system. In most cases Government support for either captured operating systems or support software should be minimal.

When the complete specification of a military family of computers is finalized, minor architectural discrepancies in the associated commercial computer family's architecture will have to be considered and resolved. Software written for machines with minor incompatibilities will either have to be adjusted accordingly, or not used.

(8) Were the results of the life cycle cost analyses meaningful?

Two life cycles cost analyses were conducted indepndently. It was hoped that an indication of the life cycle costs resulting from choosing each of the three finalists would provide a clear-cut ranking on which to base the final Committee recommendation. If so, the models used and the assumptions made in those models might significantly affect the final choice.

The life cycle cost analyses did not yield a conclusive ranking of the three finalists, however. The difference between the contenders was within the expected noise of the models. Consequently, the results dictated no clear choice for the final selection. In making its final recommendation, the Committee used the life cycle cost analyses as only one data point among others (quantitative measures score, licensing costs, Test Program results, support software capture, vendor cooperativeness, observed weaknesses in each architecture). It would be incorrect to conclude that the life cycle cost results weighed more heavily in the decision than any other available data.

The CFA Committee was well aware of the statistical closeness of the life cycle cost results. Indeed, an effort was made to determine those parameters to which the models were most sensitive, to perturbate those parameters, and to observe the effects.

While detailed descriptions of both life cycle cost models are beyond the scope of this presentation (see Volume VI), we can report the observations made by the Committee with respect to those models. One of the parameters to which the models are quite sensitive is the assumed ratio of software costs to hardware costs for tactical systems in any year. This ratio includes hardware costs for computers, main memory, secondary memory, and I/O, and the costs of applications software. The original analysis assumed a software to hardware ratio of 2:1. Other ratios tried were 1:1, 1:2, and 1:4. The results were that above 2:1 the IBM 370 was favored, at 1:1 and 1:2 the PDP-11 was favored, and below 1:4 the Interdata 8/32 was favored.

A second manipulated parameter was the yearly investment by the military to reduce identified support software deficiencies. Originally, a figure of $2 million was used. Additional calculations at the rate of $1 million/year and $3 million/year were made. The results indicated that for less than $1 million/year the IBM S/370 improves its position in life cycle cost; at $2 million/year the PDP-11 looks best and above $3 million/year the Interdata 8/32 leads.

These results reflect the Test Program and Support Software Analysis results on which they depend. In particular, the Interdata 8/32 has the most efficient architecture but is sorely lacking in support software. The IBM S/370 has the best support software position but the least efficient architecture. Thus, one would expect that if investment in support software were high and/or the ratio of software costs to hardware costs were low, then the Interdata 8/32 would look best. On the other hand, low investment in support software and/or a high software to hardware cost ratio would make the IBM S/370 look best. The middle ground is held by the PDP-11 and reflects the Committee's best estimate for these parameters. As a result, the PDP-11 ranked first on the life cycle cost models. Another point worth noting is that under the assumptions of software/hardware ratios which reflected the Committee's best estimate of the parameters, the PDP-11 had the lowest costs for between 11 to 14 of 15 Army applications considered in the bottom-up life cycle cost analysis. This is based on a projection through 1985 and is a function of the software to hardware cost ratio. This demonstrated the range of applicability of the PDP-11.

The CFA Committee could not identify any other parameters whose value was in doubt which would significantly impact the results of this analysis.

The fact that the final decision of the Committee reflected the life cycle cost results should not be interpreted as an exclusive cause-and-effect relationship.

Although the Bottom-Up model tended to corroborate the results of the Top-Down model, some may wish to criticize the former analysis on the grounds that its sources of data were fifteen Army systems and, as such, may not accurately reflect Navy needs. However, the Committee did not consider this likely.

In any case, it should be remembered that the life cycle cost analyses were not the only factor, and probably not the dominant factor, in the Committee's final decision.

(9) Was consideration given to the ease of using a computer in distributed or federated systems?

The efficiency and ease of use of the I/O structure of a computer is the primary characteristic determining the ability of a computer to be used in distributed or federated processing configurations. The importance of distributed and federated systems was clearly recognized by the CFA Committee. The quantitative criteria and benchmark programs dealing with I/O structure were consistently given higher "value" weightings than any other attribute.

(10) Why have just three (3) finalists?

If more finalists had been allowed, one of the other candidates might have shown itself superior to the others in the Test Program analysis, even though the quantitative evaluation criteria did not predict this.

Resources were limited but the Committee decided it had sufficient resources to test three candidates in depth. Even so, it proved difficult to complete even the required about 100 individual replications of the twelve test programs at the top of the Committee's prioritized list of tests for the three finalists.

Perhaps just as significant is the fact that the two of the three finalists were the only candidates judged to have passed the absolute selection criteria. Additionally, all three were grouped together at the top of the quantitative evaluation criteria.

(11) Why was diagnostic software not considered in the software capture analysis?

Diagnostic software was not considered in the software capture analysis because the hardware implementations of the militarized CFA machines will be entirely different from the commercial machines. Hence, most fault isolation type diagnostic software is unusable. On the other hand functional instruction diagnostics should run on any implementation, including militarized versions. All three final candidates have such diagnostics, but attempting to evaluate this quality appeared to be quite difficult, and we did not attempt to evaluate the quality of any other software.

(12) Why weren't such factors as the quality of the system documentation and the size of the trained labor pool for each candidate evaluated?

Since software development costs are so labor intensive, analyses of this kind might be appropriate for identifying the most cost-effective choice of architectures. Unfortunately, it is very difficult to obtain meaningful quantifiable data for such factors and to relate it to costs. This is the reason the CFA Committee did not directly evaluate such aspects.

When the Committee was screening the original list of candidate architectures, two of the seventeen quantitative criteria used dealt with the size of the existing hardware base for each architecture. By implication, this says something about the extent of trained personnel familiar with each architecture and about the

maturity of system documentation. Not surprisingly, the IBM 370 and PDP-11 scored highest on those measures.

(13) The Committee recommended a 16-bit architecture. Aren't there inherent advantages to a 32-bit architecture?

Intuitively, many people feel that a 16-bit architecture should be more efficient with respect to core storage, and that a 32-bit architecture should offer more power. The analyses done by the Committee, including the Test Programs, indicate that whereas there may be some validity to these intuitive feelings, on balance the strengths and weaknesses of the two architectures (16-bit and 32-bit) sum (or average) to the same results. In particular, the test program results indicate that one of the 32-bit architectures tested, the 8/32, was at least as efficient as the 16-bit PDP-11 architecture, and both of them tested superior to the other 32-bit architecture, the S/370. Thus, word length alone is not the determining factor. This agrees with a different kind of intuition, namely that if there were inherent advantages to one architecture, the various vendors would have gravitated towards it; instead, there are 16, 32, 36, 48, and 60 bit architectures still being produced and purchased by proud vendors and happy customers.

One must be careful to remember that the physical implementation often differs from the abstract architecture: e.g., the IBM 370 (a "32-bit" architecture) is implemented in physical bus widths of 16, 32, and 64 bits, allowing a wide performance range while maintaining a compatible architecture.

The differences in 16 and 32-bit architecture are most apparent in two areas: (1) the immediate address reach, i.e., the main memory that can be addressed without manipulation of the memory mappping mechanism, and (2) integer arithmetic. The 16-bit architecture must remap more frequently to reach a very large physical space, and usually must resort to extended-precision arithmetic to calculate 32 bit results. In return, bit traffic for smaller tasks is reduced because of shorter operands and instruction words. Based on the test results, the Committee concluded that there was no clear preference for either architecture for the general class of problems.

4. SUMMARY

In the foregoing discussion many questions considered by the CFA Committee were presented. There remains one final question which has not been formally posed to the Committee but which holds some general interest.

> If you could do the evaluation and selection again, what changes, if any, would you make?

Hindsight is usually near 20/20; i.e., the solutions always seem obvious after they are discovered. Asking what changes the Committee might wish to make, now that the experience has been acquired, is like asking how confident it is in the process and product. Even more important, it lays the foundation for further efforts in computer standardization, compatibility, and software transferability. There is no reason why the project should be <u>repeated</u>; but there are reasons why it <u>might</u> be <u>continued</u>, using what has been learned already.

Systems Analysis (as done by the Rand Corporation for DOD) has several rules of thumb. Some of those are quoted herein, and the Committee's observance of them is discussed.

1) "Treat uncertainties explicitly."

In any significant systems analysis, there are elements of the problem that defy explicit description, let alone accurate quantification. The system analyst recognizes this pitfall, and makes an effort to identify the possible impact of these unknowns.

In the CFA project, an identified uncertainty was the cooperation of the vendors -- i.e., the legal and financial aspect of entering into a contract with DOD for the use of an architecture, and its associated support software.

This contractual question was not dealt with explicitly until three finalists were selected via other criteria. The result was that the willingness (or lack thereof) of the three corporations to contract a licensing agreement with the government played a catalytic role in the decision process of many Committee members. By use of an RFP, this kind of screening could have been done early. The Committee could then have focused its limited resources on architectures whose vendors were willing to deal with DOD.

2) "Never exclude alternatives without analysis."

The strength and value of systems analysis is that the assumptions, techniques, etc., are explicit, and can be repeated, if necessary, by others, with such modifications as may seem appropriate in light of new data (or simply different circumstances).

The CFA Committee excluded several candidate architectures without <u>explicit</u> analysis. True, informal analyses were done (usually by individuals), but an explicit analysis that can be audited was not done for any architectures of CDC, Honeywell, Xerox, etc. This was in accordance with the original premise that all "good" architectures would be nominated by someone.

Given the charter of the CFA Committee (viz. to evaluate architectures, not systems), and considering the resources available to the Committee, perhaps this is the best that can be done. But if such an effort were to be continued, it would be worthwhile to approach the questions from a different perspective: Namely, to devise a uniform, broad, initial screening process that would minimize the risk of accepting any architecture that would later fail to satisfy all the criteria -- even at the risk of excluding some candidate(s) that might have been satisfactory. Such a process should also leave an audit trail. In particular, such a process should gain an early commitment from the vendor(s) involved to license the DOD to use the architecture and supporting software. Further, some group of tests should measure the performance of systems actually implemented. Later analysis in detail would determine why finalists did well in early tests, and so lead to selection of the best architecture, independent of implementation.

3) "Partial answers to relevant questions are more useful than full answers to empty questions."

Under the charter of the CFA Committee, the emphasis was on architectures, not systems. Thus, if architecture "A" is superior to "B," but system "B" is superior to "A", then the Committee would recommend adoption of "A," anyhow. In this, there is the implicit notion that DOD will continue to produce tactical software in assembly language, or that DOD will pay for the redevelopment of a host of support software, e.g., compilers for high level languages. If this is true, it is unfortunate.

Somewhere in the DOD, someone must meld the results of the CFA analysis with those of the "DOD-1" project, to point towards a superior common system. And this chosen system will be the best mixture of architecture, hardware implementation, and support software -- though it may well NOT be the best in any one of these.

The quoted rules for systems analysis were taken from [QuadE 68]. It is worth noting that the other guidelines were observed carefully, even though no one explicitly pointed them out to the Committee.

REFERENCES

[AlexB 73] Alexander, B. L., "All Applications Digital Computer (AADC) Application Studies for Advanced Light Attack Aircraft," Vought Aeronautics LTV Aerospace Corporation, March 1973.

[ButlW 72] Butler, W. I. and Kohler, G. A., "Future Naval Aircraft Subsystem/AADC Interface Definition for Operational & OBC Requirements," Final Report, Volume I, Grumman Aerospace Corporation, Report No. D-545, Contract N62269-72-C-0065, April 17, 1972.

[HopkJ 75] "DOD Weapon Systems Software Management Study," Draft, The Johns Hopkins University, Applied Physics Laboratory, May 1975.

[KilpP 73] Kilpatrick, P. S., Marshall, W. C., and Scales, E. A., "All Semiconductor Distributed Aerospace Processor/Memory Study," Volume 1: Avionics Processing Requirements, Honeywell, Inc., Air Force Avionics Laboratory Technical Report AFAL-TR-73-226, August, 1973.

[ScieC 72] "USN/USMC Future Data Systems Requirements and Their Impact on the All Applications Digital Computer (AADC)," Computer Sciences Corporation, San Diego, California, 1972.

[ApgaJ 75] Apgar, J. M., Barricelli, F. S., Coleman, A. H., Light, W. R., Jr., Morris, D. S., "Survey of ECOM Tactical Computer-Based System ADP Requirements," Communications/Automatic Data Processing Laboratory Center for Tactical Computer Sciences, U. S. Army Electronics Command, Research and Development Technical Report ECOM-4328, June 1975.

[PalmB 76] Palmer, B. S., "*Statistical Interpretation* of Selection Criteria," Naval Surface Weapons Center Memorandum DF-44:BSP:jcr, dated 5 March 1976.

[ThomM 76] Thomas, M. A., "Candidate Architecture Ranking Procedure for Army/Navy Computer Family," Naval Surface Weapons Center Memorandum DK-30:MAT:prs, dated 2 March 1976.

[StonH 76] Stone, H. S., "An Audit of the Selection Criteria for Computer Family Architecture," University of Massachusetts, Amherst, Massachusetts, 1976.

[QuadE 68] <u>Systems Analysis and Policy Planning Applications in Defense</u>, edited by E. S. Quade and W. I. Boucher, the Rand Corp., Santa Monica, California, published by American Elsevier, New York; copyright 1968, the Rand Corp.

78